Search control for CSP example

Let \((pr_1,?),..., (pr_o,?)\) be the open leafs in the current state and let
\[\text{const}(X_j) = |\{C_i \mid C_i \in C, C_i = R_i(X_{i,1},...,X_{i,k}), X_j \in \{X_{i,1},...,X_{i,k}\}\}|\]
For a problem \(pr = (x_1,..,x_n)\) let
\[\text{Csolved}(pr) = |\{C_i \mid C_i \in C, x_1,..,x_n \text{ fulfills } C_i\}|\]
Then our search control \(K\) selects the leaf to work on and the transition to this leaf (there are several possible, i.e. special case on “Less formally (II)) as follows:
If one of the $pr_j$s is solved, perform the transition that changes its sol-entry. If there are several, select one of them randomly.

Else if one of the $pr_j$s is unsolvable, perform the transition that changes its sol-entry. If there are several, again select one of them randomly.

Else

- select the leaf $(pr_j, ?)$ such that 
  a) $Csolved(pr_j) = \max_{pr_l}(\{Csolved(pr_l)\})$ 
  b) if there are several, select the deepest leaf in the tree with this property. 
  c) if there are still several, select the one the most left in the tree (tiebreaker without knowledge)
for the transition select the one with 
\(\text{Altern}(pr_j, pr_{j1},...,pr_{jk})\) such that the variable \(X_i\) we use 
to create the element in \(\text{Altern}\) is the one with 
maximal \(\text{Const-value}\). 
If there are several of those, use the one with minimal 
index \(i\) (tiebreaker without knowledge)